IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Wolrich, et al.	Art Unit	: 2187
Serial No.	: 10/780,330	Examiner	: Thammavong, Prasith
Filed	: February 17, 2004		
Title	: MEMORY MAPPING IN A PROCESSOR HAVING MULTIPLE PROGRAMMABLE UNITS		

Appeal Brief

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

(i) Real party in interest

This case is assigned to Intel Corporation, who is the real party in interest.

(ii) Related appeals and interferences

A previous notice of appeal was filed in the present case on 7/1/2009. A pre-appeal brief review resulted in withdrawal of the rejection. No appeal brief was filed and no decision was rendered by the Board.

(iii) Status of claims

Claims 36-43 are pending in the case, stand rejected, and are being appealed.

Claims 1-35 and 44-45 are cancelled.

(iv) Status of amendments

An amendment cancelling claims 28-35 and 44-45 was filed after the final rejection mailed 12/17/2009. The advisory action mailed 3/9/2010 stated the amendments had been entered.

(v) Summary of claimed subject matter

Independent claim 36 is directed to a method that includes mapping addresses in a single address space (e.g., "address space" shown in FIG. 4) to resources (e.g., locations in registers 78, 80, 76b in FIG. 2; page 10, lines 12-15) within a set of multiple programmable units (e.g., micro-engines 22a-22f and CPU 20 in FIG. 1; page 3, lines 21-22; page 4, lines 3-6) integrated within a processor (e.g., 12 in FIG 1; page 3, line 14). The single address space includes addresses for different ones of the resources in different ones of the multiple programmable units (e.g., FIG. 4 depicts address for register 78, 80, 76b locations in micro-engines 22a-22f). The method further includes providing data access to a resource within a first of the multiple programmable units to a second one of the multiple programmable units in response to a data access request (e.g., page 12, lines 21-25) of the second one of the multiple programmable units specifying an address within the single address space. There is a one-to-one correspondence between respective addresses in the single address space and respective resources within the multiple programmable units (see, e.g., FIG. 4 where an address in the address space identifies register 78, 80, 76 locations in micro-engines 22).

(vi) Grounds of rejection to be reviewed on appeal

Whether claims 36-43 were properly rejected under 35 U.S.C. s. 103(a) over Tremblay (U.S. Pat. 6,212,604) in view of Sharma (U.S. Pat. 6,055,605).

(vii) Arguments

A. Rejection under 35 U.S.C. s. 103(a) over Tremblay (U.S. Pat. 6,212,604) in view of Sharma (U.S. Pat. 6,055,605)

1. Claims 36-43

Claim 36 recites "mapping addresses in a single address space to resources within a set of multiple programmable units" where the single address space includes "addresses for different ones of the resources in different ones of the multiple programmable units". In particular, there is a "one-to-one correspondence between respective addresses in the single address space and respective resources within the multiple programmable units". Claim 36 stands rejected as obvious over Tremblay (U.S. 6,212,604) in view of Sharma (U.S. 6,055,605).

The Advisory Action (mailed 3/9/2010) equates Tremblay's processors P1 and P2 with the recited "multiple programmable units". The action seemingly equates Tremblay's registers 306 and 312 with the recited "resources within in the multiple programmable units". In Tremblay, P1 and P2 both access the same instructions in a shared instruction cache 212. However, a register address of an instruction will identify different registers depending on whether P1 or P2 execute the instruction. For example, for a given instruction address, P1 will determine a register within P1, while P2 will determine a register within P2 for the same instruction address (col. 4, lines 6-29). The precise register location determined will vary based on the setting of a register index base for each processor, but this does not alter the fact that the **same** instruction address maps to different registers - a register in P1 **and** a register in P2. This is simply not the same thing as a "one-to-one correspondence between respective addresses in the single address space and respective resources within the multiple programmable units" as recited by claim 36.

Further, Attorney for Applicant, to the extent understood, disagrees with the reasoning presented by the Examiner. For example, the Advisory Action states that "the Tremblay reference teaches the use of a single address space as the registers within the processors are accessing locations in the instruction cache 212 which allows access to the main memory 202". Attorney for Applicant does not agree with this statement nor does Attorney for Applicant understand its relevance. That is, neither instruction cache 212 nor main memory 202 are resources within P1 or P2. Nor does Attorney for Applicant understand registers 306, 312 to access the instruction cache. Instead the registers are accessed in the course of execution of instructions by P1 and P2.

.

Conclusion

In view of the foregoing, favorable reconsideration and withdrawal of the rejections is respectfully requested. If there are any questions regarding the present application, the Examiner and/or Board is invited to contact the undersigned attorney at the telephone number listed below.

Respectfully submitted,

7/15/2010
Date

/Robert A. Greenberg/
Robert A. Greenberg
Reg. No. 44,133
(978) 553-2060

(viii) Claims appendix

36. A method, comprising:

mapping addresses in a single address space to resources within a set of multiple programmable units integrated within a processor, the single address space including addresses for different ones of the resources in different ones of the multiple programmable units; and

providing data access to a resource within a first of the multiple programmable units to a second one of the multiple programmable units in response to a data access request of the second one of the multiple programmable units specifying an address within the single address space, wherein there is a one-to-one correspondence between respective addresses in the single address space and respective resources within the multiple programmable units.

37. The method of claim 36, further comprising receiving a command specifying the address in the single address space.

38. The method of claim 37, wherein the command comprises one selected from the following group: a read command and a write command.

39. The method of claim 37, wherein the receiving the command comprises receiving the command from a programmable processor.

40. The method of claim 39, wherein the programmable processor comprises a programmable processor integrated within the processor; and

wherein the multiple programmable units comprise multiple programmable engines and the programmable processor.

41. The method of claim 36, wherein the resources within the set of multiple programmable units comprises register locations within the multiple programmable units.

42. The method of claim 36, wherein the single address space comprises addresses corresponding to shared resources external to the multiple programmable units.

43. The method of claim 36, wherein the multiple programmable units comprise multiple programmable multi-threaded units.

(ix) Evidence appendix

None.

(x) Related proceedings appendix

None.